Application No.: 09/726,779	§	
Filed: November 29, 2000	§	Examiner: Pillai, Namitha
Inventor(s):	§	Group/Art Unit: 2173
Chris Cifra, Kevin Schultz, Jeff	§	Atty. Dkt. No: 5150-43700
Kellam, Jeff Correll, Nicolas	§	
Vazquez, And Christophe	§	
Caltagirone	§	
Title: AUTOMATIC	§	
GENERATION OF	§	
PROGRAMS WITH GUI	§	
CONTROLS FOR	§	
INTERACTIVELY	§	
SETTING OR VIEWING	§	
VALUES		

REPLY BRIEF

Sir or Madam:

Further to the Examiner's Answer of July 28, 2006, Appellant presents this Reply Brief, and respectfully requests that this Reply Brief be considered by the Board of Patent Appeals and Interferences.

STATUS OF CLAIMS

      Claims 1, 3-14, 16-27, 29-40, and 42-59 are currently pending in the application. Claims 1, 3-5, 7-14, 16-18, 20-27, 29-31, 33-40, and 42-59 stand rejected under 35 U.S.C. § 102(e) over U.S. Patent No. 6,298,474 B1 to Blowers et al. ("Blowers"), and claims 6, 19, and 32 stand rejected under 35 U.S.C. § 103(a) over Blowers, and are the subject of this appeal.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 3-5, 7-14, 16-18, 20-27, 29-31, 33-40, and 42-59 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over U.S. Patent No. 6,298,474 B1 to Blowers et al.

Claims 6, 19, and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,298,474 B1 to Blowers et al.

ARGUMENTS

The below is presented in response to the Examiner's Response to Argument in the Examiner's Answer. Appellant respectfully notes that the Examiner's Answer substantially repeats arguments made earlier in the prosecution, which have been addressed in the Appeal Brief. The Examiner's Answer further includes quite a bit of additional description or characterization of Blowers's invention that Appellant respectfully submits is not germane to the patentability of the present claims, and in which Appellant was unable to discern new arguments. Appellant has presented below responses to any *new* arguments in the Examiner's Answer directed to the independent claims. Thus, where no new discernable arguments were presented in the Examiner's Answer, Appellant has not added further responses, but relied on those presented in the Appeal Brief.

Section 102(e) Rejections

Claims 1, 11, 14, 24, 27, 37, 43, 47, 53, 54, 55, 57, 58, 59

Claims 1, 11, 14, 24, 27, 37, 43, 47, 53, 54, 55, 57, 58, and 59 are separately patentable because the cited reference does not teach or suggest the limitations recited in these claims.

In the Response to Arguments, the Examiner continues to argue that Blowers's machine vision system and the components used for developing the program (including the task sequencing engine 46) are separate, but then states, "The environment in which the program is developed includes **the task sequence engine coupled with the runtime interface both responsible for creating and executing the program** (Figure 3 and column 9, lines 13-18)." (*emphasis added*) The Examiner further asserts that "The machine vision system in Figure 2 is a separate system, which is associated with the program, created but is separate from the program and is independently executed from the machine vision system (column 8, lines 28-34) [*sic*]."

Appellant notes that the Examiner herself has admitted that the task sequence engine is part of both the prototyping/development environment and the runtime

environment—specifically, that the task sequence engine (46) is responsible for both *creating* **and** *executing* the program.

As Appellant discussed previously in the Appeal Brief at p.10:13 – 11:5, Blowers teaches a design engine or task sequencer engine 46 that is used to configure a desired sequence of functional tasks and create a program implementing the sequence, and that is also used to execute the program. More specifically, a user creates the desired sequence by selecting graphical representations or icons from the tool boxes of FIG. 5. Once the desired sequence has been created, it is stored in a condensed method within an inspection sequence file 52 which is useable by the engine 46. The engine 46 takes the condensed stored sequence from the file 52 and executes it through the runtime screen of FIG. 9 (See Col. 8, line 61 – Col. 9, line 25). In other words, Blowers teaches that the engine 46 is used to configure a desired sequence of functional tasks, and the desired sequence is then executed by the engine 46. Blowers is quite clear that engine 46 is used to both develop the program and to execute the program.

Appellant further notes that the cited Figure 2 clearly shows that the machine vision system includes the host computer, which includes the prototyping (development) environment (e.g., including the task sequencer engine 46). Additionally, Figure 3 clearly shows that the task sequencer engine 46 is used (for development) via the task sequencer interface 50, and (for runtime execution) via the runtime interface 54. In other words, as Figures 2 and 3 make clear, the development system and the runtime system (i.e., the machine vision system) are *not* separate and independent systems, but rather, as particularly shown in Figure 3, are a single integrated system. Thus, the Examiner's assertion that the machine vision system is separate and distinct from the components used for developing the program is incorrect. Moreover, contrary to the Examiner's assertions, Blowers's program is *not* operable to execute independently of the prototyping application, since the engine 46 is required for creating *and* for executing the program.

Similarly, since the engine 46 is used for both the creation and execution of the program, Blowers does not, and cannot, teach "wherein said automatically generating the graphical user interface comprises automatically creating one or more graphical user interface elements associated with the one or more parameters of the first functional

operation, wherein during execution of the program, at least one of the one or more graphical user interface elements is displayed and is operable to receive user input *independently of the prototyping application*", as claimed.

The Examiner attempts to equate Blowers's results window with Appellant's claimed automatically generated graphical user interface comprising automatically created graphical user interface elements. However, Appellant notes that, as the Examiner herself states, Blowers's results user interface is automatically generated during execution of the associated program, *not*, during creation of the program. More specifically, the results user interface is simply a window in Blowers's GUI that displays results from the program execution, and is thus more properly characterized as a part of the development/execution environment, *not* part of the program. Additionally, the results user interface is clearly only for displaying results, and is nowhere described as including one or more graphical user interface elements that are displayed and operable to receive user input *independently of the prototyping application*, as claimed.

Appellant thus maintains that Blowers's results user interface is neither automatically generated as part of the program, nor does the interface include graphical user interface elements that are displayed and operable to receive user input *independently of the prototyping application*.

The Examiner further argues that Blowers's tree structure is created and executed using the task sequence engine of Figure 3, and that this "allows for the program to be executed independently from the machine vision system of Figure 2. However, as Appellant explained above, the system of Figure 3 is included in the system of Figure 2 (e.g., is resident on the host computer 28). Note, for example, that the system of Figure 3 includes the task sequencer engine 46, as well as the runtime interface 54, results engine 56, and results interface 60, as well as hardware manager 40, camera / I/O /calibration registry 44, and I/O camera frame grabber 43, all used in the machine vision system. Moreover, Appellant notes that the Abstract of Blowers specifically characterizes Blowers's invention thusly:

> A method, a system and a computer-readable storage medium having stored therein a program for interactively developing **a graphical control-flow structure and associated application software for use in a machine vision system** is provided. (*emphasis added*)

Thus, it is clear that the program is not executed independently from the machine vision system. Furthermore, contrary to the Examiner's assertion that Blowers's tree structure is executed by the task sequence engine, per Blowers, the (manually created) tree structure *represents* the program, and is used to generate the program, but is not itself the program, and more specifically, is *not* executed.

Thus, Blowers fails to teach or suggest all the features and limitations of claim 1, and so Appellant respectfully submits that the teachings of Blowers clearly do not meet the standard for anticipation which requires that the <u>identical</u> invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, <u>arranged as in the claim</u>. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1457, 221 USPQ 481, 485 (Fed. Cir. 1984). Appellants' invention as recited in claim 1 is clearly not anticipated by Blowers, and so claim 1 is allowable.

<u>Claims 40</u>

Claim 40 is separately patentable because the cited reference does not teach or suggest the limitations recited in this claim.

The Examiner again argues that Blowers's development environment and machine vision system are separate and distinct, and that the task sequence engine of Figure 3 creates and executes the program *independently* of the machine vision system of Figure 2. However, as argued at length above with respect to claim 1, Blowers's ask sequence engine 46 is used in both the development of the program, and in the execution of the program as part of the machine vision system. Appellant respectfully notes that the entire point of executing the program is to operate the machine vision system. In fact, Blowers col.8:28 – 33 reads thusly:

> Referring now to FIG. 3, there is illustrated in block diagram form various software and hardware components for interactively developing a graphical, control-flow structure and associated application software **for**

use in the machine vision system 20 of FIG. 2 using the computer system of FIG. 1... (*emphasis added*)

Clearly, the program is executed by the engine 46 as part of the machine vision system, and so is not executed independently of the machine vision system. As discussed above, the system of Figure 3 is included in the system of Figure 2 (e.g., is resident on the host computer 28). Note, for example, that the system of Figure 3 includes the task sequencer engine 46, as well as the runtime interface 54, results engine 56, and results interface 60, as well as hardware manager 40, camera / I/O /calibration registry 44, and I/O camera frame grabber 43, all used in the machine vision system.

Thus, Blowers fails to teach all the features and limitations of claim 40, and so claim 40 is allowable..

Claim 48, 50, 51

Claims 48, 50, and 51 are separately patentable because the cited reference does not teach or suggest the limitations recited in these claims.

The Examiner again argues that Blowers's tree structure is the same as Appellant's automatically generated graphical program, citing the connected nodes of the tree structure. Appellant notes that, as argued previously, the task sequencer list of Figure 6, i.e., the tree structure of Figure 6, is not generated automatically, but rather via manual user-selection of icons from tool boxes. Nowhere does Blowers teach or suggest generating such a tree structure (or a graphical program) automatically. Additionally, the Examiner continues to suggest that the Blowers's tree structure is executable, which is simply not the case. As Blowers makes clear in Col. 8, line 61 – Col. 9, line 25, a user creates the desired sequence by selecting graphical representations or icons from the tool boxes of FIG. 5. Once the desired sequence has been created, it is stored in a condensed method within an inspection sequence file 52 which is useable by the engine 46. The engine 46 takes the condensed stored sequence from the file 52 and executes it through the runtime screen of FIG. 9. Clearly, the tree structure is constructed manually by the user (via input selecting the various nodes representing functions and adding them to the tree structure), after which the condensed method is generated, stored, and subsequently

retrieved and executed by the engine 46. Thus, the tree structure is neither automatically generated, nor executed.

Thus, Blowers fails to teach all the features and limitations of claim 48, and so claim 48 is allowable.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1, 3-14, 16-27, 29-40, and 42-59 was erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5150-43700/JCH.

Respectfully submitted,

    /Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35,198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: September 28, 2006 JCH/MSW